Area of interest based interaction and GeoVisualization with WebGL

Bektas, Kenan ; Cöltekin, Arzu

Abstract: In this paper we present an experimental approach and a prototype implementation to manage level of detail in geovisualizations. The approach is based on identifying the area of interest of the user (mouse location or gaze point) and removing perceptually irrelevant detail in the visual field. Prototype implementation achieves this in real time using WebGL. By utilizing this test environment we want to systematically reveal the potential of integrating gaze based interaction modalities to the mobile webmapping applications.

In: Proceedings of The Graphical Web Conference 2012.
September 11-14 2012 Zurich, Switzerland

Area of Interest Based Interaction and GeoVisualization with WebGL

Kenan Bektas, Arzu Coltekin
{Kenan.Bektas | Arzu.Coltekin} (at) geo.uzh.ch
Department of Geography, University of Zurich, Switzerland

Abstract

In this paper we present an experimental approach and a prototype implementation to manage level of detail in geovisualizations. The approach is based on identifying the area of interest of the user (mouse location or gaze point) and removing perceptually irrelevant detail in the visual field. Prototype implementation achieves this in real time using WebGL. By utilizing this test environment we want to systematically reveal the potential of integrating gaze based interaction modalities to the mobile webmapping applications.

Introduction

Based on the marketed mobile devices and the applications that accompany them, we observe that there is a clear demand for ubiquitous location based services with the widespread use of the Internet and the smart phones. This demand is met with enthusiastic supply: we have so much data that it is difficult to process or distribute it quickly and meaningfully. This relatively new 'data overload' calls for testing new approaches to manage the level of detail for transmission and visualization. In this paper we present an experimental approach and a prototype implementation to manage level of detail in geovisualizations. The approach is based on identifying the area of interest of the user (mouse location or gaze point) and removing perceptually irrelevant detail in the visual field. Prototype implementation achieves this in real time using WebGL and paves the way for further testing.

The human eyes do not process visual information in a uniform fashion (e.g., Marr, 1982, Geisler and Perry, 1999). Certain psychophysical properties of the human eye (coupled with the mechanisms that control attention) determine the limits of our visual perception, including the way we perceive resolution and color. The color sensitive photoreceptor cells, the so called cones, are highly concentrated on the foveal region and almost half of the primary visual cortex (i.e. V1) is dedicated to processing signals transmitted from this particular region on retina (Ware, 2004). This simply leads to the fact that, in contrast to the uniform resolution displays, we perceive the scenes with significantly higher resolution at the point of gaze than in the peripheral region. Inspired from this, foveation has been developed as an image coding and visualization technique with the objective of addressing technical and perceptual issues in reducing information intensity (Burt, 1988). In that sense, foveation is a one-to-one mapping of fovea's structure on to any visualization. When a display

is rendered based on the idea that we see the world in a non-uniform fashion, it is called a 'gaze-contingent display' (GCD) (Perry and Geisler, 2002). Note that not all GCDs are foveated, i.e. the rendering may be contingent to the gaze but the resolution may change based on a random non-uniform model, not necessarily based on foveation. Ideally, a foveated GCD leads to perceptually lossless image coding (Çöltekin, 2009), and can be used for visualization and analysis of multi-layered raster and vector types of geo-data (Bektas and Çöltekin, 2011). We call this approach geofoveation.

Framework

Geofoveation is an experimental concept and is currently being tested for offline and online environments. The approach has great potential for online use because of its aggressive yet 'perceptually lossless' data reduction ability (Çöltekin, 2009). To implement web based geofoveation, WebGL appears to be a promising tool. WebGL is a recently developed JavaScript API that allows developers to access graphical processing unit (GPU) on the client side and to control the graphics rendering pipeline. It employs dedicated graphics hardware resources for the computationally intensive visualization tasks (e.g. information visualization or online games) that are performed on a web browser (Williams, 2011). It is supported by many modern web browsers (e.g. Chrome, Firefox, Opera, Safari). While WebGL offers possibilities to make geofoveation computationally more efficient, to test the gaze-contingency paradigm, we need to have a software module that collects and communicates the gaze point data (Figure 1). For this purpose, we utilize the Text 2.0 Framework (Biedert et al., 2010). The framework has a distributed architecture and includes a tracking server, diagnosis interface and the core Text 2.0 plug-in.

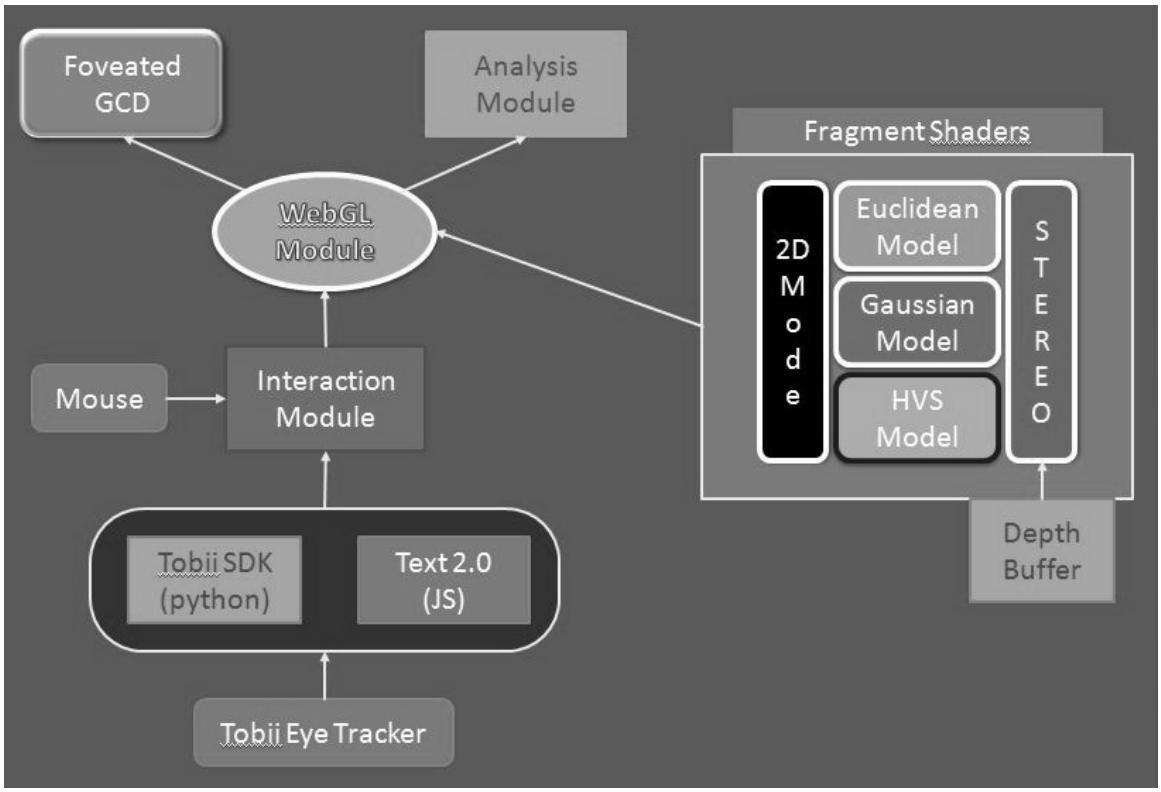

Figure 1: Framework showing the modules of the test environment.

As demonstrated in Figure 1, after initiating the server and connecting to the eye tracker, the necessary gaze information can be acquired within any html code, by utilizing the Text 2.0 event handlers which are very similar to the traditional mouse handlers used in javascript. The software development kit (SDK) distributed by Tobii (i.e. Tobii SDK 3.0) enables the developers to implement applications that use the functionalities of the Tobii eye tracking devices. The SDK includes libraries written in .NET, Python, C++ and Objective C languages. Therefore, the application developers may easily utilize this SDK for an alternative implementation.

Vision Model

In this paper we present the alpha version of our test environment. The foveated image coding is implemented as a fragment shader. Once the user's area of interest is determined (e.g. gaze point or mouse pointer), the scene is foveated accordingly. That is, the detail is removed from the periphery based on a model of the foveal vision. In this environment, various visual models can be tested. The models can be as simplistic i.e. based on the Euclidean distance (see example code below) or a 2D Gaussian distribution, more sophisticated which make use of the contrast sensitivity of the human eye i.e. a stereoscopic or a monoscopic Human Visual System (HVS) Model (Figure 1). Below a code snippet is presented to demonstrate the basic workings of the shader.

Sample code:

```
<script id='fshader' type='x-shader'>

precision mediump float;

uniform sampler2D u_image; // texture1: Background

uniform sampler2D u_image_lr; // texture2: Foreground

varying vec2 v_texCoord;

uniform vec2 center;

void main() {

float dist = (distance(v_texCoord, center)  * (1.0));

vec4 color =  texture2D(u_image, v_texCoord, dist) + texture2D(u_image_lr, v_texCoord, dist);

gl_FragColor = color;

}

</script>
```

Data

In typical geovisualization applications, the input data may have multiple layers and can be in raster or vector formats (see Figure 2). In our test-environment, so far we experimented with raster data (e.g. jpg, .png, .tiff), however the concept and the framework are not limited by the data format. In a typical scenario the background can be an aerial image of a city which is overlaid by some foreground (vector) layers including street network, bus stops or prominent landmarks. In our implementation we simulate these additional data layers by pre-processed raster images which are aligned to the background. The output can be considered as a composition of some images from the same region of interest. As the number of layers increase the visual complexity of resulting visualization also increases. In our implementation we read the visual data from images and create mipmaped textures. In the fragment shader we sample each pixel of the resulting image from the mipmap levels. The pixels which have a larger Euclidean distance to the point of interest (i.e. center) are selected from the higher level of the mipmaps which in turn have a coarser resolution (Figure 3). Therefore, the information load on foveated visualizations is less than those which have the original resolution.

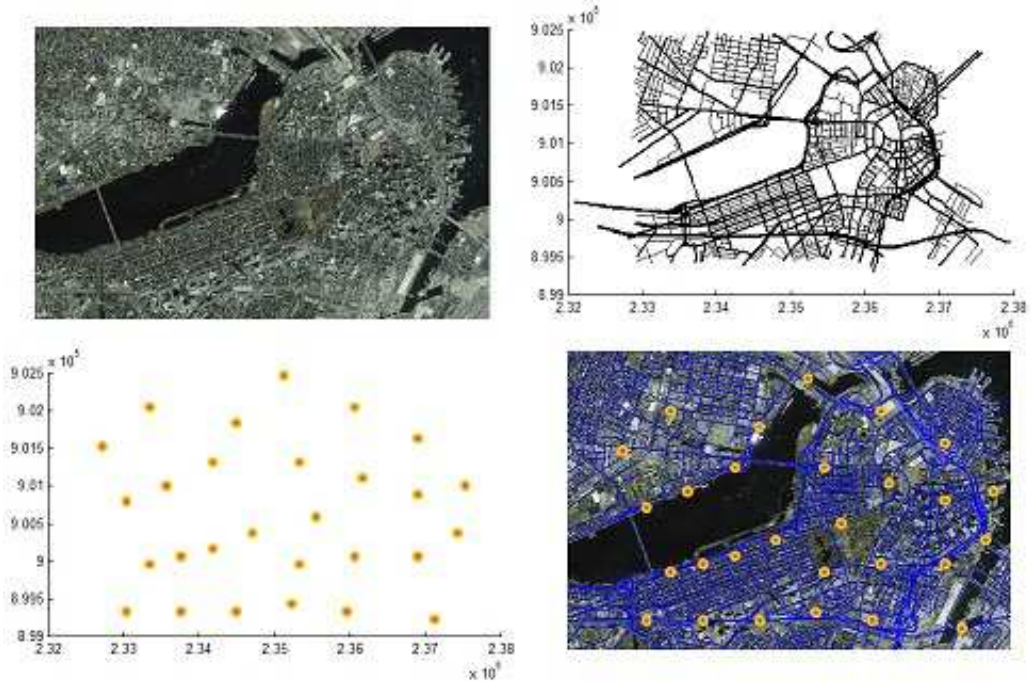

Figure2: Data for a typical Geovisualization application

Conclusions

This short paper describes a work in progress. We have tested our current implementation on Firefox 15.0, with one background (aerial image boston.jpg) and one foreground layer (busstops.jpg). The images both have a resolution of (1024X1024). We currently have implemented two interaction

modalities: the system mouse and the Tobii X120 eye tracker and create seamless foveated visualization for both layers in real-time (Figure 3).

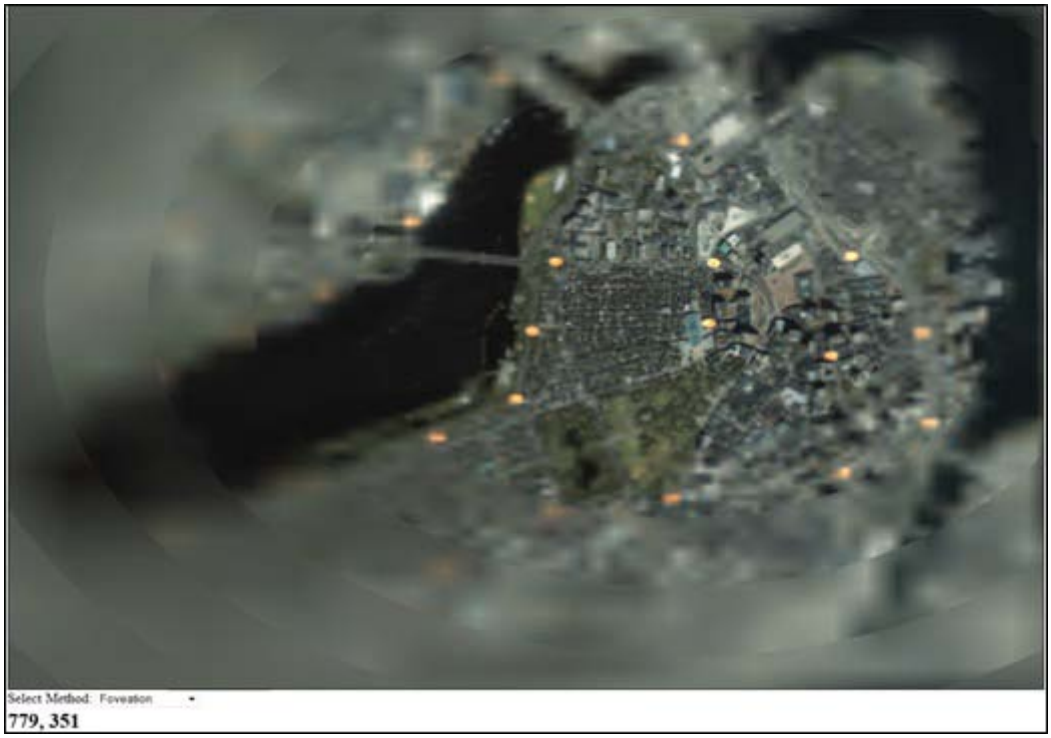

Figure3: A screenshot from the geofoveated visualization of two layers on a web browser.

Processing images which have a non-power-of-two (NPOT) resolution is still an issue in WebGL. In order to process these we have used (a well known solution) the skewed versions of the input images (in horizontal dimension) and visualized these in a canvas which has the original image resolution. As a future work we plan to prepare multi-layered data sets to test various geofoveated visualization scenarios. By utilizing this test environment we want to systematically reveal the potential of integrating gaze based interaction modalities to the mobile webmapping applications. We hope to contribute to a) managing the visualization of large volumes of geographic data online and b) employ experimental user studies for the analysis of visual information load on geovisualizations in a web based environment.

Acknowledgements

This research is funded by Swiss National Science Foundation (SNF award number 200021_120434/1). We would like to thank GeoEye (www.geoeye.com) and Office of Geographic Information (MassGIS), Commonwealth of Massachusetts Information Technology Division for their cooperation and providing the data.

References

Bektas K and Çöltekin A. An Approach to Modeling Spatial Perception for Geovisualization. *Procedia Social and Behavioral Sciences*. Volume 21: 2011; p. 53-62.

Biedert R, Buscher G, Schwarz S, Möller M, Dengel A and Lottermann T. The Text 2.0 Framework - Writing Web-Based Gaze-Controlled Real-time Applications Quickly and Easily. In: *Proceedings of the International Workshop on Eye Gaze in Intelligent Human Machine Interaction* (EGIHMI). 2010; p. 114-117

Burt PJ. Attention Mechanisms for Vision in a Dynamic World. In: *Proceedings of the 9th International Conference on Pattern Recognition*. 1988; p. 977-987

Coltekin A. Space-Variant Image Coding for Stereoscopic Media. In: *Proceedings of the IEEE Picture Coding Symposium*. 2009; p. 1-4

Geisler W and Perry JS. Variable-resolution displays for visual communication and simulation. *The Society for Information Display*. Volume 30: 1999; p. 420-423

Marr, D. *Vision: A Computational Approach*. Freeman & Co., San Francisco, 1982

Perry JS and Geisler W. Gaze-contingent Real-time Simulation of Arbitrary Visual Fields. In: *Proceedings of the SPIE, Human Vision and Electronic Imaging* VII, Volume 4662: 2002; p. 57-69

Ware C. The Environments, Optics, Resolution, and the Display. In: *Information Visualization: Perception for Design*. 2nd ed. San Francisco, Morgan Kaufman; 2004, Chapter 2, p. 29-68

Williams LJ. L*earning HTML5 Game Programming: A Hands-on Guide to Building Online Games Using Canvas, SVG, and WebGL*. Addison Wesley 2011